Principles of Machine Learning Systems

2: Model Compression

Nicholas D. Lane
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines
Roadmap for Today

Source: Bianco

Source: OpenAI
Roadmap for Today

Source: Han

http://mlsys.cst.cam.ac.uk/teach

Principles of Machine Learning Systems – v1.5
Roadmap for Today

Model Compression

- Pruning
  - Pruning Weights
  - Pruning Neurons
  - Pruning Blocks
  - Pruning Heads & Layers
- Quantization
- Knowledge Distillation
  - Distillation Architectures
  - Collaborative Learning
  - Multiple Teachers
  - Adversarial Methods
  - Distilling Transformers
- Parameter Sharing
  - Character-aware LMs
  - For Embedding Matrix
  - For Transformers
- Tensor Decomposition
  - Two Low-Rank Factors
  - Block Diagonal Factors
  - TT and Block Term Decomposition
- Linear Complexity
  - Star Transformer
  - Linformer
  - Sparse Sinkhorn Transformer
  - Efficient Attention
  - Linear Transformers

Source: Gupta
Roadmap for Today

Architecture (Re-)Design

1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression

3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Matrix Decomposition

Bottleneck: Repeated Matrix Multiplications

\[ \begin{align*}
\text{Total Operations: } & m \times n \times 1 \\
\text{Total Operations: } & m \times k \times 1 + k \times n \times 1
\end{align*} \]

Gain in memory and computations when:

\[ k < \frac{m \times n}{m + n} \]
Matrix Decomposition

\[ x^{L+1} \rightarrow W^L \rightarrow x^L \]

New Inserted Layer

\[ x^{L+1} \rightarrow U \rightarrow V \rightarrow x^L \]

\[ m \times k \quad k \times n \quad n \times 1 \]
Decomposition Benefits

Google SpeakerID Model (FC Layers)

32 KB
ARM Cortex M3

16 KB
ARM Cortex M0

2-4% degradation in accuracy
CNN Kernel Decomposition

Inception v3
(2014)

5x5 filter

Apply sequentially

5x1 filter

1x5 filter

separable filters

VGG
(2014)

5x5 filter

decompose

Two 3x3 filters

Apply sequentially

Source: V. Sze

Replacing a large filter with a series of small filters

AlexNet (2012)
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Efficient Architecture Evolution

Source: Lee

http://mlsys.cst.cam.ac.uk/teach
MobileNet Example

Object Detection
Photo by Juaneic (CC BY 2.0)

Face Attributes
Google Doodle by Sarah Harrison

Finegrain Classification
Photo by HarshLight (CC BY 2.0)

Landmark Recognition
Photo by Sharon VanderKsay (CC BY 2.0)

Source: Google
# MobileNet Block

<table>
<thead>
<tr>
<th>Type / Stride</th>
<th>Filter Shape</th>
<th>Input Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv / s2</td>
<td>$3 \times 3 \times 3 \times 32$</td>
<td>$224 \times 224 \times 3$</td>
</tr>
<tr>
<td>Conv dw / s1</td>
<td>$3 \times 3 \times 32$</td>
<td>$112 \times 112 \times 32$</td>
</tr>
<tr>
<td>Conv / s1</td>
<td>$1 \times 1 \times 32 \times 64$</td>
<td>$112 \times 112 \times 32$</td>
</tr>
<tr>
<td>Conv dw / s2</td>
<td>$3 \times 3 \times 64$</td>
<td>$112 \times 112 \times 64$</td>
</tr>
<tr>
<td>Conv / s1</td>
<td>$1 \times 1 \times 64 \times 128$</td>
<td>$56 \times 56 \times 128$</td>
</tr>
<tr>
<td>Conv dw / s1</td>
<td>$3 \times 3 \times 128$</td>
<td>$56 \times 56 \times 128$</td>
</tr>
<tr>
<td>Conv / s1</td>
<td>$1 \times 1 \times 128 \times 256$</td>
<td>$28 \times 28 \times 256$</td>
</tr>
<tr>
<td>Conv dw / s1</td>
<td>$3 \times 3 \times 256$</td>
<td>$28 \times 28 \times 256$</td>
</tr>
<tr>
<td>Conv / s1</td>
<td>$1 \times 1 \times 256 \times 512$</td>
<td>$14 \times 14 \times 256$</td>
</tr>
<tr>
<td>Conv dw / s2</td>
<td>$3 \times 3 \times 512$</td>
<td>$14 \times 14 \times 512$</td>
</tr>
<tr>
<td>Conv / s1</td>
<td>$1 \times 1 \times 512 \times 1024$</td>
<td>$7 \times 7 \times 512$</td>
</tr>
<tr>
<td>Conv dw / s2</td>
<td>$3 \times 3 \times 1024$</td>
<td>$7 \times 7 \times 1024$</td>
</tr>
<tr>
<td>Conv / s1</td>
<td>$1 \times 1 \times 1024 \times 1024$</td>
<td>$7 \times 7 \times 1024$</td>
</tr>
<tr>
<td>Avg Pool / s1</td>
<td>Pool $7 \times 7$</td>
<td>$7 \times 7 \times 1024$</td>
</tr>
<tr>
<td>FC / s1</td>
<td>$1024 \times 1000$</td>
<td>$1 \times 1 \times 1024$</td>
</tr>
<tr>
<td>Softmax / s1</td>
<td>Classifier</td>
<td>$1 \times 1 \times 1000$</td>
</tr>
</tbody>
</table>

Original Independent Kernels

One for each kernel

Source: Howard
MobileNet Benefits

<table>
<thead>
<tr>
<th>Model</th>
<th>ImageNet Accuracy</th>
<th>Million Mult-Adds</th>
<th>Million Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 MobileNet-224</td>
<td>70.6%</td>
<td>569</td>
<td>4.2</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>69.8%</td>
<td>1550</td>
<td>6.8</td>
</tr>
<tr>
<td>VGG 16</td>
<td>71.5%</td>
<td>15300</td>
<td>138</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>ImageNet Accuracy</th>
<th>Million Mult-Adds</th>
<th>Million Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv MobileNet</td>
<td>71.7%</td>
<td>4866</td>
<td>29.3</td>
</tr>
<tr>
<td>MobileNet</td>
<td>70.6%</td>
<td>569</td>
<td>4.2</td>
</tr>
</tbody>
</table>

Parameter Reduction

\[
\frac{\text{Depthwise Separable}}{\text{Regular Conv}} = \frac{M(N + DK^2)}{D_k^2 \times M \times N} = \frac{1}{D_k^2} + \frac{1}{N}
\]

MACs Reduction

\[
\frac{\text{Depthwise Separable}}{\text{Regular Conv}} = \frac{D_k^2 \times M \times (N + DK^2)}{D_k^2 \times M \times D_F^2 \times N} = \frac{1}{D_k^2} + \frac{1}{N}
\]

Source: Howard
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. **Parameter and Channel Pruning**
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Pruning Justification

Source: Han

http://mlsys.cst.cam.ac.uk/teach

Principles of Machine Learning Systems – v1.5
Related Natural Phenomena

50 Trillion Synapses → 1000 Trillion Synapses → 500 Trillion Synapses

- Newborn
- 1 year old
- Adolescent

Source: Han
Pruning Algorithm

- Range of pruning criteria
- Example: Prune when absolute weight is < a threshold

Source: Koul
Alternative Pruning Criteria

- Magnitude
- Energy
  - Estimation
  - Measurement
- Random
- etc.

Source: V. Sze
Pruning Schedule

Source: Han
## Pruning Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Unpruned</th>
<th>Prune Ratio</th>
<th>Fine-tuned</th>
<th>Scratch-E</th>
<th>Scratch-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>VGG-19</td>
<td>93.50 (±0.11)</td>
<td>30%</td>
<td>93.71 (±0.09)</td>
<td>93.31 (±0.26)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>80%</td>
<td>93.71 (±0.08)</td>
<td>93.64 (±0.09)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95%</td>
<td>93.34 (±0.13)</td>
<td>93.21 (±0.17)</td>
<td>93.63 (±0.18)</td>
</tr>
<tr>
<td></td>
<td>PreResNet-110</td>
<td>95.04 (±0.15)</td>
<td>30%</td>
<td>95.06 (±0.05)</td>
<td>94.84 (±0.07)</td>
<td>95.11 (±0.09)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>80%</td>
<td>94.55 (±0.11)</td>
<td>93.76 (±0.10)</td>
<td>94.52 (±0.13)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95%</td>
<td>92.35 (±0.20)</td>
<td>91.23 (±0.11)</td>
<td>91.55 (±0.34)</td>
</tr>
<tr>
<td></td>
<td>DenseNet-BC-100</td>
<td>95.24 (±0.17)</td>
<td>30%</td>
<td>95.21 (±0.17)</td>
<td>95.22 (±0.18)</td>
<td>95.23 (±0.14)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>80%</td>
<td>95.04 (±0.15)</td>
<td>94.42 (±0.12)</td>
<td>95.12 (±0.04)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95%</td>
<td>94.19 (±0.15)</td>
<td>92.91 (±0.22)</td>
<td>93.44 (±0.19)</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>VGG-19</td>
<td>71.70 (±0.31)</td>
<td>30%</td>
<td>71.96 (±0.36)</td>
<td>72.81 (±0.31)</td>
<td>73.30 (±0.25)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50%</td>
<td>71.85 (±0.30)</td>
<td>73.12 (±0.36)</td>
<td>73.77 (±0.23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95%</td>
<td>70.22 (±0.38)</td>
<td>70.88 (±0.35)</td>
<td>72.08 (±0.15)</td>
</tr>
<tr>
<td></td>
<td>PreResNet-110</td>
<td>76.96 (±0.34)</td>
<td>30%</td>
<td>76.88 (±0.31)</td>
<td>76.36 (±0.26)</td>
<td>76.96 (±0.31)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50%</td>
<td>76.60 (±0.36)</td>
<td>75.45 (±0.23)</td>
<td>76.42 (±0.39)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95%</td>
<td>68.55 (±0.51)</td>
<td>68.13 (±0.64)</td>
<td>68.99 (±0.32)</td>
</tr>
<tr>
<td></td>
<td>DenseNet-BC-100</td>
<td>77.59 (±0.19)</td>
<td>30%</td>
<td>77.23 (±0.05)</td>
<td>77.58 (±0.25)</td>
<td>77.97 (±0.31)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50%</td>
<td>77.41 (±0.14)</td>
<td>77.65 (±0.09)</td>
<td>77.80 (±0.23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95%</td>
<td>73.67 (±0.03)</td>
<td>71.47 (±0.46)</td>
<td>72.57 (±0.37)</td>
</tr>
<tr>
<td>ImageNet</td>
<td>VGG-16</td>
<td>73.37</td>
<td>30%</td>
<td>73.68</td>
<td>72.75</td>
<td>74.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60%</td>
<td>73.63</td>
<td>71.50</td>
<td>73.42</td>
</tr>
<tr>
<td></td>
<td>ResNet-50</td>
<td>76.15</td>
<td>30%</td>
<td>76.06</td>
<td>74.77</td>
<td>75.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60%</td>
<td>76.09</td>
<td>73.69</td>
<td>74.91</td>
</tr>
</tbody>
</table>

Source: Wang

---

**Accuracy Loss**

- Pruning + Quantization
- Pruning Only
- Quantization Only
- SVD

**Model Size Ratio after Compression**

- 2% to 20%
- 0.5% to 5%
- 0.5% to 11%
- 0.5% to 14%
- 0.5% to 17%
- 0.5% to 20%
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. **Parameter Quantization**
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Guess who
Guess who
Numerical Representation

Floating-point Numbers (Decimal)

32-bit Single-Precision Floating-point Number

64-bit Double-Precision Floating-point Number
Low Precision Training

Source: Gupta

http://mlsys.cst.cam.ac.uk/teach  
Principles of Machine Learning Systems – v1.5  
27
Quantization Implementation

• Linear vs. Non-linear
  • Linear – representable space is divided equally “X”
  • Non-linear – representable space is divided un-equally “O”

• Round to nearest vs. Stochastic rounding
  • Round to nearest – each number is represented by the closest representable number
  • Stochastic rounding – each number is represented by the closest higher or lower value with equal probabilities
Training Quantized Models
Binary: Extreme Quantization

Space Reduction 32x

Compute Reduction No FP Ops!

Source: Geiger
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Knowledge Distillation

Distillation of a large model into a small one is performed in a teacher-student setup whereby the student network is trained to replicate the teacher’s raw output (logits).
Knowledge Distillation

• Knowledge distillation does not require actual data for the student’s training since the goal is for it to match the teacher’s output not the implied labels.

• Thus, students can be trained on random inputs just as well as on the original data! The teacher provides the targets in real time.
Knowledge Distillation

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>SST-2</th>
<th>QQP</th>
<th>MNLI-m</th>
<th>MNLI-mm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc</td>
<td>F1/Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td>1</td>
<td>BERTLARGE (Devlin et al., 2018)</td>
<td>94.9</td>
<td>72.1/89.3</td>
<td>86.7</td>
<td>85.9</td>
</tr>
<tr>
<td>2</td>
<td>BERTBASE (Devlin et al., 2018)</td>
<td>93.5</td>
<td>71.2/89.2</td>
<td>84.6</td>
<td>83.4</td>
</tr>
<tr>
<td>3</td>
<td>OpenAI GPT (Radford et al., 2018)</td>
<td>91.3</td>
<td>70.3/88.5</td>
<td>82.1</td>
<td>81.4</td>
</tr>
<tr>
<td>4</td>
<td>BERT ELMo baseline (Devlin et al., 2018)</td>
<td>90.4</td>
<td>64.8/84.7</td>
<td>76.4</td>
<td>76.1</td>
</tr>
<tr>
<td>5</td>
<td>GLUE ELMo baseline (Wang et al., 2018)</td>
<td>90.4</td>
<td>63.1/84.3</td>
<td>74.1</td>
<td>74.5</td>
</tr>
<tr>
<td>6</td>
<td>Distilled BiLSTM_{SOFT}</td>
<td>90.7</td>
<td>68.2/88.1</td>
<td>73.0</td>
<td>72.6</td>
</tr>
<tr>
<td>7</td>
<td>BiLSTM (our implementation)</td>
<td>86.7</td>
<td>63.7/86.2</td>
<td>68.7</td>
<td>68.3</td>
</tr>
<tr>
<td>8</td>
<td>BiLSTM (reported by GLUE)</td>
<td>85.9</td>
<td>61.4/81.7</td>
<td>70.3</td>
<td>70.8</td>
</tr>
<tr>
<td>9</td>
<td>BiLSTM (reported by other papers)</td>
<td>87.6†</td>
<td>–/82.6‡</td>
<td>66.9*</td>
<td>66.9*</td>
</tr>
</tbody>
</table>

Source: Tang

Achieve comparable results with ELMo, while using roughly **100 times** fewer parameters and **15 times** less inference time.
Roadmap for Today

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach
Principles of Machine Learning Systems – v1.5
Basic Compression Pipeline

Pruning: less number of weights
- Train Connectivity
- Prune Connections
- Train Weights

Quantization: less bits per weight
- Cluster the Weights
- Generate Code Book
- Quantize the Weights with Code Book
- Retrain Code Book

Huffman Encoding
- Encode Weights
- Encode Index

Source: Han

http://mlsys.cst.cam.ac.uk/teach

Principles of Machine Learning Systems – v1.5
Resource-aware Pipeline

- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget.
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture).
- Requires **very few hyperparameters** to tune.

![Diagram of Resource-aware Pipeline]

*Source: Yang*

---

http://mlsys.cst.cam.ac.uk/teach

Principles of Machine Learning Systems – v1.5
Summary of the Day

Architecture (Re-)Design
1. Early Evolution; Decomposition as a tool
2. Efficient Architectures; Example: MobileNet

Architecture Compression
3. Parameter and Channel Pruning
4. Parameter Quantization
5. Knowledge Distillation
6. Compression pipelines

http://mlsys.cst.cam.ac.uk/teach